Amendments to the Specification:

Please replace the paragraph at page 1, lines 13-22 with the following amended paragraph:

Figure 1 of the accompanying drawings shows a computer system including a typical communications bus architecture. A plurality of modules are connected to a combined read and write bus 1 and to a separate control bus 2, both of which are well known tri-state buses. The modules may be, for example, memory devices, graphics controllers, CPU's, and so on. The control bus and the read/write bus service all the requirements of the system, enabling the modules to transfer data between one another or externally, for example to external memory or control devices.

Please replace the paragraph at page 17, lines 19-24 with the following amended paragraph:

Each module M1 to M5 in column 28 is assigned a relative priority level in column 29. In the example shown in Figure 14, modules M1 and M4 are assigned a high priority level, modules M2 and M5 a medium priority level and module M3 a low priority level. The priority levels are stored in the priority level storage means 27 in the arbitration unit 21.

Please replace the paragraph at page 17, lines 25-31 with the following amended paragraph:

The arbitration scheme starts at step A. The stack positions are shown at 30 in Figures 15, 16, 17, 18 and 19. The initial set up of the arbitration scheme is to arrange the modules into initial stack positions (step [[A]] B) which are stored in the stack storage means 26. These initial positions are illustrated in Figure 15. It will be appreciated that the initial stack positions are arbitrarily chosen. In the example shown, M1 is at the top of the stack and M5 at the bottom.

Please replace the paragraph spanning page 17, line 32 through page 18, line 14 with the following amended paragraph:

In step [[B]] C, the arbitration unit 21 receives respective transaction requests from any number of the modules M1 to M5. For example, all five modules may wish to make

transaction requests at the same time. The transaction requests are received by the request receive means 25 and are forwarded to the control means 24. At step C, the control means 24 determines which of the transaction requests are associated with the highest priority level of the modules issuing requests. In the example where all five modules M1 to M5 issue transaction requests, modules M1 and M4 can be seen to have the highest priority levels of the requesting modules. In step D, the control means obtains the highest priority transaction request. The control means then masks out (step E) all requests from lower priority modules. Thus, only the requests from modules M1 and M4 are processed further. This is illustrated in Figure 16.

Please replace the paragraph at page 21, lines 23-30 with the following amended paragraph:

Since the two arbiters 32 and 33 operate independently, the transaction bus can be fully utilised, by using the vacant clock cycle associated with a write data transaction to issue a read transaction on the transaction bus. This is illustrated in Figure 21. The transaction bus 35 is ideally alternated between read and write states so that the write data bus 36 is also fully utilised carrying the two packet write data packets.

Please replace the paragraph spanning page 22, line 32 through page 23, line 13, with the following amended paragraph:

Arbitration for the read bus 37 can be similar to that for the write bus 36 but can be simplified if each master module is able to accept the matured read data as soon as it is made available. A similar two-stage filter and stack arbitration system can be used to arbitrate between the various sources of read response data, and no interaction is required from the write or transaction buses. The read data transfer is achieved over two cycles (two half packets). Once again, the priorities could be programmable, but in a preferred example, the priorities are fixed. When mature data is ready at a target, that target module indicates to the read arbiter that it wishes to send data to a particular master module. Use of the read bus is then controlled by the retirement arbiter.

Please replace the paragraph at page 23, lines 15-26, with the following amended paragraph:

An alternative arbitration unit 39 includes separate write bus and read bus arbiters 40, 41, for example as illustrated in Figure 22, connected to the bus by control lines 42 in the same way as for Figure 20. In such a case, the bus preferably includes two transaction buses- one for read and one for write. Such a system is illustrated in Figure 23 where the transaction bus TW 43 is associated with the write data bus W 44 and the read transaction bus TR 46 is associated with the read data bus R 45. As illustrated in Figure 24, since the read and write data buses 44 and 45 and the read and write transaction buses 43 and 46 are entirely separate, then it is possible to perform simultaneous independent read and write operations.

Please replace the paragraph spanning page 23, line 28 through page 24, line 10, with the following amended paragraph:

As described above, in a computer architecture employing a bus system and multiple modules connected to that bus system, some form of arbitration is required to determine which of the modules can have access to the bus system. Preferably, the computer system is defined by a memory map in which respective address ranges 56 and 57 are allocated to individual modules. In such a system, each module can address the other modules simply by using a single address value. Thus, if module M3 wishes to write data to a particular address, it simply issues address data equivalent to that address. This address data needs to be decoded to determine the target module identity. Preferably, each module M1, M2 and M3 supplies information to all the other modules indicating when it is busy (i.e. unavailable for transactions).

Please replace the paragraph spanning page 27, line 19 through page 28, line 5, with the following amended paragraph:

Figure 28 shows representationally a master module MN, which is connected to a bus 70, and which provides a module function 79. The master module MN requests data using a transaction request inducing transaction address data supplied on the transaction bus by a transaction output request stage 74. The transaction request also includes a transaction sequence tag which is produced by a sequence producer counter 71 and sent to the bus as shown at 73. This sequence tag indicates the relative order in which the transaction has been

produced. When read data is received, via an input 76, that read data packet has a read packet sequence tag associated with it which is received by an input 77. The read sequence tag, which is equivalent to the transaction sequence tag output by the master with its transaction request, is used to indicate where the read data packet should be stored within a two port memory, ie. RAM, buffer 75. The read data packet 76 is input via the memory write port and is written at a position within the memory indicated by the sequence tag 77.

Please replace the paragraph at page 29, lines 6-18, with the following amended paragraph:

The first in first out FIFO register 83 supplies address data as shown at 84 to the target function 86, whilst identification and sequence data is passed to a further first in first out FIFO buffer 87. This buffer 87 provides a tag queue, and is used so that master module identification and sequence data can be recombined with data packets read by the target function 86. When a data packet to be output to the master is returned by the target function 86, then the associated tag (ID and sequence data) is output onto the bus at the same time, thereby identifying the output read data. The tags are output using control lines 88 and the read data packet output on the line 89.